IMPLEMENTACIÓN DE FILTRADO DE PAQUETES (PF) CON OPENBSD EN DISPOSITIVOS (HARDWARE) DE BAJOS RECURSOS TECNOLÓGICOS, COMO PRIMER BASTIÓN DE LA SEGURIDAD INFORMÁTICA.

Sepúlveda Manrique, Luis Jesús
ljsepulveda@gmail.com
Universidad Piloto de Colombia

Abstract **—** This document proposes the implementation of a packet filtering as a primary parameter in information security and using free software "OpenBSD" in devices with low technological benefits and low monetary cost, with packet filtering being an efficient technology that does not require the use of hardware with high technological configurations at the time of implementation in homes and companies. The filtering of packages is an alternative to be the first bastion in computer security in data networks, to minimize risks of computer attacks.

Resumen —Con este documento se propone la implementación de un filtrado de paquetes como parámetro primordial en la seguridad de la información, y empleando el software libre "*OpenBSD*" en dispositivos con bajas prestaciones tecnológicas y bajo costo monetario, siendo el filtrado de paquetes una tecnología eficiente que no precisa del empleo de hardware con altas configuraciones tecnológicas al momento de la implementación en hogares y empresas. El filtrado de paquetes es una alternativa para ser el primer bastión en la seguridad informática en las redes de datos, con la finalidad de minimizar riesgos de ataques informáticos.

Palabras clave **—**OpenBSD, Filtrado de paquetes, Firewall, reuso tecnológico, BeagleBone Black.

I. INTRODUCCIÓN

En este documento se realiza un aproximación al empleo de sistemas de filtrado de paquetes como parte fundamental de la implementación de medidas de seguridad informática para hogares y empresas. A fin de minimizar costos de operación o gastos adicionales en la puesta de funcionamiento de este tipo de medidas de seguridad, se realiza el planteamiento de utilizar dispositivos de reuso o de reacondicionamiento tecnológico, tomando como referencia la ley 1672 de 2013 del Congreso de la República [1], donde define y plantea el uso de este tipo de equipos. Un caso de éxito en el empleo de equipos en desuso es el programa "*Computadores para Educar*", con más de un millón de equipos entregados [2]; así mismo, se propone como alternativa el uso de dispositivos embebidos para cumplir la tarea de filtrado de paquetes, equipos que no superan los 70 dólares.

Como parte fundamental de esta propuesta se sugiere el uso de software libre OpenBSD, el cual posee altos estándares de seguridad y criptografía, que lo hace ideal para la finalidad propuesta. En el documento se plantea una serie de conceptos que abarcan el tema de firewall, teoría de filtrado de paquetes y la instalación, configuración y uso de OpenBSD pf (packet filter), como sistema para el filtrado de paquetes, para lo cual se propone una configuración con el empleo de dos tarjetas de red una para el acceso a red de internet y otra para la red interna LAN.

II. DISPOSITIVOS DE BAJOS RECURSOS TECNOLÓGICOS

Dentro de este documento se hace referencia a la instalación de un sistema de filtrado de paquetes para dispositivos de bajos recursos tecnológicos, entre ellos se encuentran equipos de cómputo obsoletos o en

desuso y dispositivos de sistemas embebidos como la plataforma de desarrollo BeagleBone Black.

A. Equipos por reacondicionamiento o reuso tecnológico.

A medida que avanza la tecnología tanto en software como en hardware muchos equipos electrónicos quedan en desuso u obsoletos, lo cual conlleva a engrosar la acumulación de desechos electrónicos, conocidos como "basura tecnológica" o "e-waste", término según la organización "raee.org.co" hace referencia a: "*aparatos dañados, descartados u obsoletos que consumen electricidad. Incluye una amplia gama de aparatos como computadores, equipos electrónicos de consumo, celulares y electrodomésticos que ya no son utilizados por sus usuarios*". [3]. Para el caso de Colombia el Congreso de la República, a través de la ley 1672 de 2013, estableció los lineamientos para el manejo de los residuos de aparatos eléctricos y electrónicos (RAEE), en el artículo 4, definiendo reacondicionamiento como: "*Procedimiento técnico de renovación en el cual se establecen las condiciones funcionales y estéticas de un aparato eléctrico y electrónico con el fin de ser usado en un nuevo ciclo de vida. Puede implicar además reparación, en caso de que el equipo posea algún daño*". También se define el concepto de reuso como: "*se refiere a cualquier utilización de un aparato o sus partes, después del primer usuario, en la misma función para la que el aparato o parte fueron diseñados*" [1]. En el marco de esta ley establece una serie de obligaciones las cuales fueron reglamentadas en el decreto No. 284 de 2018, en la que destaca la obligación de los usuarios. En el artículo 2.2.7A.2.3, numeral primero se enuncia que: "*Prevenir la generación de los RAEE mediante prácticas para la extensión de la vida útil de los AEE*" [4], con lo cual, y acorde a la ley, los ciudadanos tenemos la obligación de aprovechar los recursos tecnológicos que por variadas circunstancias van quedando obsoletos.

Un ejemplo de este tipo de disposición de elementos tecnológicos es la iniciativa de "*Computadores para Educar*" [2], la cual se inició con la recolección de equipos de cómputo en desuso u obsoletos, los cuales fueron reacondicionados para ser reutilizados en entidades educativas y sociales en el territorio nacional, las estadísticas de esta organización refieren la reutilización y puesta en funcionamiento de más de 1.886.755 equipos entregados [2]. En este documento presento la propuesta y como parte de las obligaciones de los usuarios y consumidores de equipos electrónicos y eléctricos, la opción de reutilizar equipos en desuso u obsoletos para la implementación de "firewall" con tecnología de filtrado de paquetes como un primer bastión de seguridad informática en las redes de datos, y empleando el software libre OpenBSD 6.3; que en términos de recursos económicos de bajo costo tanto para hogares y empresas no supera $ 300.000 pesos (COP) de inversión. En caso de comprar un equipo de cómputo usado, o de cero costo para el caso de la reutilización de equipos; este software precisa el uso de recursos mínimos o capacidades reducidas de hardware como son:

- Procesador Intel Pentium III, Atom, AMD Athlon, entre otros con velocidad mínima de 500 Mhz. Memoria RAM de 512 Mb
- Disco duros de 2 Gb
- Dos (2) interfaces o tarjetas de red.

B. Dispositivo BeagleBone Black.

Creado con base en la plataforma de desarrollo BeagleBoard, que inició el desarrollo de sistemas embebidos, estructurados y con filosofía de software y hardware abierto, dichos sistemas son desarrollados y mantenidos por la organización sin ánimo de lucro "*BeagleBoard Foundation"* [5] de la cual hace parte Texas Intruments, fabricante de la familia de procesadores AM335X Cortex A8 ARM, procesador que se encuentra en la plataforma de desarrollo de la referencia BeagleBone Black (figura 1), definido por el como "*una plataforma de desarrollo de bajo costo*" [5]. El costo promedio del dispositivo según el fabricante es de 49 dólares.

Fig. 1. BeagleBone Black.

Fig. 1: Se realiza comparativo del tamaño del dispositivo BeagleBone Black. Fuente: el autor.

Dentro de las características básicas de conectividad de la BeagleBone Black se encuentran las siguientes:

- Puerto USB para alimentación y comunicaciones
- Puerto USB
- Puerto para micro SD
- Puerto Ethernet RJ45
- Puerto HDMI para video
- Dos (2) headers de desarrollo cada uno con 46 pines.

En la tabla 1 se aprecian las principales características de la BeagleBone Black.

TABLA 1

Características Técnicas BeagleBone Black

Ítem.	**Características**	
Procesador	1Gz Sitara AM3359AZCZ100	
Motor gráfico	SGX530 3D, 20M Polygons/S	
Memoria SDRAM	512 MB DDR3L 400Mhz	
Flash	2GB,8bit Embedded MMC	
PMIC	TPS65217C PMIC regulator and one additional LDO.	
Alimentación	miniUSB o Jack DC	5VC Vía header de expansión
PCB	3.4” x 2.1”, 6 capas	
Indicadores	1 Poder, 2 Ethernet, 4 Leds para control del usuario	
Puerto serial	UART0 vía pin 6 3.3V TTL.	
Ethernet	10/100, RJ45	
SD/Conector MMC	microSD, 3.3V	
Entradas usuario	Botón reset, botón boot y botón poder	
Salida de video	16b HDMI, 1280×1024 (MAX)1024×768, 1280×720, 1440x900w/Soporte EDID	
Audio	Vía HDMI, Estéreo	
Conectores de expansión	Poder 5V, 3.3V VDD_ADC(1.8V)3.3V I/OMcASP0, SPI1, I2C, GPIO(69), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8 MAX) , 4 Timers, 4 Puertos Seriales, CAN0, EHRPWM(0.2) , Interrupciones XDMA, Botón de Poder	
Consumo	210-460mA @ 5V Dependiendo de la actividad y velocidad del procesador.	
Peso	39.68 gramos	

Tabla 1: Se detallan las características del dispositivo. Fuente: https://electronilab.co/tienda/beaglebone-black-arm-cortex-a8-1ghz/

De la tabla 1 se destacan las características de procesamiento y rendimiento de la BeagleBone Black, entre las cuales se encuentran:

- Procesador Sitara AM335x Cortex ARM con velocidad de 1Ghz.
- Memoria RAM de 512 MB tipo DDR3.
- Almacenamiento flash a bordo eMMC de 2GB y 8 bits.
- Acelerador de gráficos 3D.

La BeagleBone Black viene con el sistema operativo de Linux Angström, sin embargo, actualmente se encuentran múltiples distribuciones Linux que pueden ser instaladas plataforma BeagleBone Black; dentro de las cuales se incluyen:

- Ubuntu
- Android
- Fedora
- Debian
- Kali Linux
- OpenBSD
- ArchLinux
- Gentoo

Por lo anterior, se aprecia que este tipo de dispositivo presenta características y soporte para la instalación del sistema operativo OpenBSD propuesto en este documento.

III. FIREWALL

En la publicación especial 800-41 "*Guidelines on firewalls and firewall policy***”** del “National Institute of Standards and Technology” (NIST), se define firewall (cortafuegos) como: ”*dispositivos o programas que controlan el flujo de tráfico de red entre redes o hosts que emplean diferentes posturas de seguridad”* [6]. En este contexto, el documento describe de manera general las diferentes tecnologías y tipos de firewall, determinando unas directrices para planear, implementar y generar políticas en cuanto la forma en que los firewall (cortafuegos) deben manejar tráfico en la red; precisando al respecto que: *“ Antes de crear una política de firewall, se debe realizar alguna forma de análisis de riesgo para desarrollar una lista de los tipos de tráfico que necesita la organización y categorizar cómo deben protegerse”* [6]

Dentro de las diferentes tecnologías de firewall acorde a lo establecido en la guía NIST 800-41 [6] se listan las siguientes:

- Filtrado de paquetes.
- Inspección con estado
- Firewall de aplicación
- Gateway proxy de aplicación

- Servidores proxy dedicados.
- Redes privadas virtuales (VPN)
- Controles de acceso a las redes
- Gestión unificada de amenazas (UTM).
- Firewall de aplicación web.
- Firewalls para infraestructuras virtuales.

Cada una de estas tecnologías de firewall se pueden y deben combinar, a fin de ofrecer una mayor protección a las conexiones de red, tanto entrantes como salientes. Se va documentar en relación a la tecnología más primitiva del firewall, como lo es el filtrado de paquetes. En la actualidad hace parte importante de la implementación de un perímetro básico de seguridad para las redes de datos en hogares o corporativas.

IV. TECNOLOGÍA DE FILTRADO DE PAQUETES (PACKET FILTERING)

Dentro de las tecnologías de cortafuegos (firewall), las más básica a implementar es el filtrado de paquetes (packet filter). Las primeras tecnologías de firewall implementadas correspondían a filtros de paquetes que se encargaban de realizar un enrutamiento [6], permitiendo el control de acceso para cada una de las direcciones host que se definían en las reglas del dispositivo; y así establecer la comunicación entre cada una de las máquinas. Fueron denominados firewall de inspección sin estado, debido que básicamente se encargan de bloquear o permitir el paso a los paquetes de datos acorde a una lista de reglas establecidas de aceptación o denegación de los paquetes [7], y según van llegando a la interfaz de red, este tipo de tecnología se implementa generalmente a nivel del sistema operativo. El filtrado de paquetes se encuentra destinado principalmente a las capas 3 (nivel de red) y capa 4 (capa de transporte) del modelo OSI, adicionalmente el filtrado de paquetes permite realizar NAT (Network Address Translation) acorde a lo establecido en el RFC1631 [8], así mismo, es posible realizar acondicionamiento y normalización del tráfico de red, permitiendo un control del ancho de banda de la red. En resumen, el filtrado de paquetes (pf) trabaja u opera a nivel de conexiones, puertos, protocolos y paquetes.

Debido a que el filtrado de paquetes se enfoca en las opciones de permitir o bloquear paquetes, en donde el protocolo encargado de enrutar los paquetes a través de internet es el protocolo IP, definido en el RFC791 "*Internet Protocol Darpa Internet Program Protocol Specification"* [9], donde, las reglas de filtrado se centran principalmente en los campos de dirección IP de origen y la dirección IP de destino, compuestas cada una por 32 bits. Para el caso de IPV4 y para IPV6, las direcciones de origen y desino están conformadas por 128 bits.

Otro de los aspectos importantes en el filtrado de paquetes es el concerniente a la capa de transporte del modelo OSI, donde toma papel importante los protocolos destinados para la entrega de los datagramas. Los más importantes a tomar en cuenta son: protocolo TCP, definido en el RFC793 "*Transmission Control Protocol Darpa Internet Program Protocol Specification*" [10]; en el cual los campos más importantes de este protocolo son los relacionados con el puerto de origen y el puerto de destino, así como lo relacionado con el campo de las banderas o "flags" (URG, ACK, PSH, RST, SYN, FIN). El otro protocolo a tener en cuenta para el filtrado de paquetes es el protocolo UDP, definido en el RFC768 "*User Datagram Protocol*" [11].

Al momento de establecer un sistema de filtrado de paquetes este debe tomar en cuenta las siguientes consideraciones:

1) Examinar cada uno de los encabezados de los protocolos y los paquetes de datos, para lo cual se definen diferentes reglas de filtrado.
2) Establecer un conjunto de reglas las cuales se establece qué hacer con el paquete de datos. Las reglas buscan la información definida en cada uno de los filtros como son: dirección IP de origen y destino, número de puerto de origen y destino, entre otros.
3) El filtrado de paquetes debe ser capaz de generar acciones a tomar a partir del resultado del examen realizado al paquete de datos, entre las que se encuentran:

- Con base en las reglas puede aceptar solo paquetes que sean seguros y rechazar todos los otros paquetes.
- Según el conjunto de reglas rechaza solo paquetes que no sean seguros y acepta todos los otros paquetes.
- Preguntar al usuario que hacer con paquetes que no están establecidos en las reglas de filtrado.
- Bloquear usuarios de una dirección IP de origen definida, cuando recibe demasiados paquetes en un corto período de tiempo.
- Permitir un sin número de reglas de filtrado y establecer cualquier tipo de acciones a tomar con los paquetes de datos, para lo cual el administrador puede crear las reglas que estime

pertinentes para tal fin, en aras de preservar la seguridad de la red.

B. Tipos de filtrado de paquetes.

Acorde con la tecnología del filtrado de paquetes se han establecido dos grandes categorías como son: filtrado de paquetes sin estado y filtrado de paquetes con estado.

1) Filtrado de paquetes sin estado [12]. Hace referencia al tipo más básico de filtrado del tráfico que pasa por la red ya sea de entrada o de salida y en la cual se aplica una lista de reglas para denegar o permitir el paquete, para este caso se realiza una inspección del paquete y se compara con las listas generadas, entre los cuales se encuentran los siguientes:

- Dirección IP de destino o dirección IP de origen.
- Puertos de enlace, siendo los más usuales 80, 22, 23, 443, entre otros
- Protocolos UDP, TCP, ICMP, etc.
- Cabecera de protocolo TCP.

Este tipo de filtrado puede permitir como modo se seguridad básica, que no se muestren las configuraciones de red, así como evitar que se acceda a protocolos o se realicen inicios de sesión desde un ataque *outsaider*. Debido al tipo de inspección realizado a los paquetes en este tipo de filtrado es propenso a recibir ataques de saturación de la red o ataques de tipo SYN *flooding*

2) Filtrado de paquetes con estado [14]. Generalmente se le conoce como filtrado dinámico, ya que opera a nivel de conexión o de flujo, se agregó el concepto de las tablas de estado donde se mantienen los registros de los diferentes inicios de sesión, cada que se genera un nuevo paquete, este es comparado o verificado con las tablas de estado para establecer si se encuentra una conexión asociado a ella, en caso de no hallarlo este paquete es descartado. Este tipo de filtrado de paquetes aumentó la seguridad de la información e inicialmente se diseñó como una protección contra los ataques de *spoofing*, los cuales eran muy comunes en el sistema de filtrado de paquetes sin estado, convirtiendo el filtrado de paquetes dinámico en un factor fundamental a la hora de generar políticas y estrategias para la defensa en profundidad [13] de las organizaciones; así mismo, en este tipo los sistemas de filtrado con estado permite reducir el gasto de recursos de cómputo lo que nos facilita la instalación en equipos de bajo costo y bajos recursos de hardware, como lo son equipos de reciclaje tecnológico o plataformas de sistemas embebidos como la BeagleBone Black entre otros dispositivos.

Al momento de especificar o definir el conjunto de reglas [14] se hace importante que se realicen como mínimo las siguientes verificaciones en los encabezados de los paquetes: La dirección IP de origen del paquete, la dirección IP del destino, el número del puerto tanto UDP como TCP, identificación del protocolo IP, los mensajes ICMP entre otros aspectos. Dentro de la implementación de un filtrado de paquetes existen dos características básicas:

- De forma predeterminada aceptar todo y denegar que pasen paquetes de forma explícita.
- De forma predeterminada denegar todo y permitir que pasen paquetes de forma explícita.

Siendo la directiva de denegar todo, la más recomendada a la hora de configurar las reglas de filtrado de paquetes, sin embargo, es necesario habilitar cada servicio con el correspondiente protocolo de comunicación.

En aras de garantizar mayor seguridad a la hora de fijar las reglas se debe aclarar los conceptos de rechazar o denegar. Rechazar implica que se dé como respuesta un mensaje de error ICMP, lo que genera que esta respuesta pueda ser usada para realizar un ataque de denegación de servicio, ya que esta respuesta incluye información útil para cualquier atacante; por ende, la acción más recomendada es la de denegar los paquetes, en esta acción simplemente se descarta el paquete sin devolver ningún tipo de respuesta.

C. Filtrado de paquetes entrantes.

1) Filtrar de dirección de origen remota. La forma de identificar al remitente de un paquete es la dirección IP de origen, sin embargo y a fin de evitar ciertos ataques como "*spoofing*" se hace necesario al momento de implementar un filtrado de paquetes con dirección de origen se realice una denegación principalmente a las siguientes de direcciones IP de origen: mi dirección IP, las direcciones IP privadas de las clases A, B y C, direcciones IP de servicios locales o de bucle, direcciones IP reservadas en la clase E y direcciones IP en clase D multidifusión.

2) Filtrar dirección destino local. Generalmente la tarjeta NIC ignora los paquetes que no tiene como destino esta tarjeta, sin embargo, una excepción a esta regla lo conforman las direcciones IP de difusión general, en la cual la dirección IP 255.255.255.255 corresponde a una dirección de destino.

3) Filtrar puerto de origen remoto. Generalmente el puerto de origen web remoto es el puerto 80 (http) o 443 (https), para los demás casos las peticiones realizadas al servidor local desde servidores remotos se asignan en puertos no privilegiados, que corresponden a los puertos del 1024 al 65535.
4) Filtrar de puerto destino local. Las peticiones realizadas desde clientes remotos generalmente establecen un puerto de destino con el número de servicio asignado específicamente, las respuestas de los servidores remotos tendrán en el puerto de destino rangos no privilegiados de 1024 al 63535.
5) Filtrar estado de conexión TCP entrante. Generalmente para este tipo de filtrado se hace referencia de los indicadores de estado del protocolo IP SYN, ACK) [14]; por tanto, cada paquete que proceda de un servidor remoto debe tener activado el indicador ACK debido a que siempre serán la respuesta a una petición realizada desde la red local.

D. Filtrado de paquetes salientes

1) Filtrar de dirección de origen local. Para el caso en que se trate de hogares o pymes en las cuales es limitado el uso de equipos el filtrado de la dirección IP de origen corresponde a la misma dirección de los equipos de los usuarios.
2) Filtrar dirección destino remota. Para el filtrado de paquetes con dirección remota se toman en cuenta dos casos principales así: filtrar paquetes con destino a clientes remotos que son direccionados desde servicios del usuario, y paquetes con destino a servidores remotos con los cuales se establecido una conexión desde el usuario.
3) Filtrar puerto de origen local. Se hace necesario y recomendado definir el puerto de servicio a emplear para las conexiones salientes teniendo claro especificar los puertos para aplicaciones del cliente y otra para las aplicaciones de los servidores locales. Para el caso de los aplicativos del cliente se debe tener precaución, ya que generalmente la salida se realiza por puertos no privilegiados, por lo que se deben asignar a puertos específicos y así aumentar la seguridad en la salida de paquetes de la red.
4) Filtrar de puerto destino remoto. Se debe realizar un filtrado para que las conexiones de los clientes solo se conecten a los puertos de servicio asociados a los aplicativos, garantizando que se asegure la red al permitir o vigilar que programas intenten acceder a los servidores de internet y minimizar la exploración de puertos.
5) Filtrar estado saliente de conexiones TCP. Acorde al saludo de tres vías del protocolo de conexión de paquetes TCP, el indicador SYN se encontrará activado en la primera petición, pero el indicador ACK no estará activo; posteriormente se activa el indicador ACK para los siguientes paquetes, por tanto, un filtrado del estado saliente cliente local solo permite un SYN o un ACK activo, para el caso de los servidores local siempre se permitirá que los paquetes que salgan tengan el indicador ACK activo.

V. SISTEMA OPERATIVO OPENBSD

El sistema operativo OpenBSD es sistema multiplataforma descendiente de UNIX y basado en NetBSD. OpenBSD es de libre distribución basado en la licencia BSD (*Berkeley Software Distribution*) desarrollada por la Universidad de Berkeley de California, la ventaja de esta licencia de software libre permite que el código fuente puede ser modificado, estudiado y distribuido sin ningún tipo de permiso. Actualmente OpenBSD se encuentra soportada por la fundación de sin ánimo de lucro "*The OpenBSD Foundation*" [15] quienes se encargan de realizar las actualizaciones y soportar el funcionamiento de este sistema operativo. Dentro de los objetivos fundamentales del sistema operativo se encuentran la portabilidad, la estandarización, exactitud, la seguridad proactiva y la criptografía integrada [16]. Respecto del objetivo fundamental de criptografía, el sistema OpenBSD incluyó en el núcleo el protocolo IPsec desde la versión 2.1 lanzada en el año de 1997, cabe destacar en este objetivo que la criptografía implementada en el sistema operativo se encuentra relaciona además con los siguientes aspectos: OpenSSH con soporte para las diferentes versiones hasta la 2.0, funciones de resumen criptográfico, generación de números aleatorios y sistema de transformaciones criptográficas [17], respecto de la seguridad OpenBSD es considerado uno de los sistemas operativos tipo UNIX más seguros que existen actualmente.

Las características respecto de la criptografía, la seguridad implementada en el sistema operativo, así como la baja cantidad de recursos tecnológicos para su instalación y funcionamiento donde requiere un mínimo de memoria RAM de 256 Mb y un espacio libre en disco de 500 Mb para la instalación, aunado con que fue el primer sistema operativo que se liberó con un sistema de filtrado de paquetes (pf), como se mencionó anteriormente el sistema operativo es multiplataforma [18], en la tabla 2 se aprecian las diferentes plataformas de instalación que se encuentran

soportadas en la versión 6.3, versión que fue librada el día 15 de abril de 2018.

TABLA 2

Plataformas soportadas por OpenBSD 6.3

alpha	Digital Alpha-based systems
amd64	AMD64-based systems
arm64	64-bit ARM systems
armv7	ARM-based devices, such as BeagleBone, BeagleBoard, PandaBoard ES, Cubox-i, SABRE Lite, Nitrogen6x and Wandboard
hppa	Hewlett-Packard Precision Architecture (PA-RISC) systems
i386	Standard PC and clones based on the Intel i386 architecture and compatible processors
landisk	IO-DATA Landisk systems (such as USL-5P) based on the SH4 cpu
loongson	Loongson 2E- and 2F-based systems, such as the Lemote Fuloong and Yeeloong, Gdium Liberty, etc.
luna88k	Omron LUNA-88K and LUNA-88K2 workstations
macppc	Apple *New World* PowerPC-based machines, from the iMac onwards
octeon	Cavium Octeon-based MIPS64 systems
sgi	SGI MIPS-based workstations
sparc64	Sun UltraSPARC and Fujitsu SPARC64 systems

Tabla 2: Plataformas soportadas para la instalación de OpenBSD 6.3. Fuente. https://www.openbsd.org/plat.html.

Por estas características entre otras más, fueron predominantes a la hora de escoger el sistema para la implementación de filtrado de paquetes basado en OpenBSD, como primer bastión en la seguridad de las redes de datos en hogares y corporativas, para ser instalado en dispositivos de obsoletos o de reuso tecnológico, así como, en la placa de desarrollo BeagleBone Black, la cual se encuentra soportada por la distribución armv7, para el caso de dispositivos obsoletos o de reuso hay aplicaciones disponibles en sistema con arquitecturas i386 o amd64.

VI. FILTRADO DE PAQUETES CON OPENBSD

Previa la instalación y configuración de PF de OpenBSD, se hace necesario conocer y comprender conceptos básicos relacionados con el funcionamiento y configuración del filtrado de paquetes en OpenBSD. En la figura 2, se aprecia el diagrama de flujo [19] correspondiente al procesamiento de un paquete bajo OpenBSD, el cual finaliza con dos opciones: permitir el paso del paquete o rechazar el paquete; posteriormente se hace referencia a una serie de conceptos y comandos relacionados con configuraciones básicas para filtrado de paquetes [20], en este tipo de sistema operativo.

Fig. 2. Diagrama de flujo análisis de paquetes

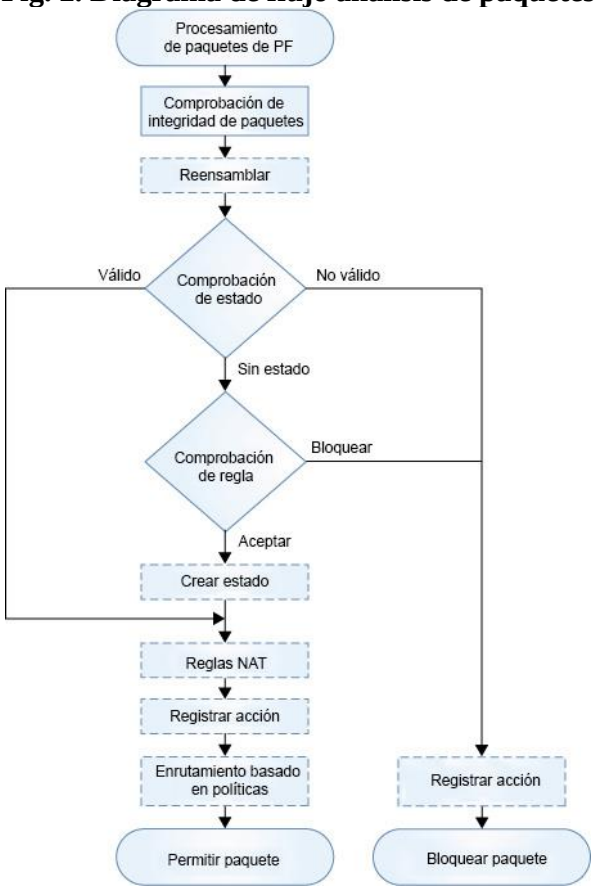

Fig. 2: Diagrama de flujo para el análisis básico de un paquete bajo el sistema operativo OpenBSD. Fuente: https://docs.oracle.com/cd/E53394_01/html/E54829/pfovw-proc.html

A. Configurar las interfaces.

Previo a realizar la implementación del sistema de filtrado de paquetes, se hace necesario realizar la configuración de cada una de las interfaces de red a emplear, y realizar pruebas de conectividad, los siguientes comandos nos permiten realizar este proceso en OpenBSD:

LAN:
echo inet [dirección IP] [mascara] > etc/hostname.[nombre]

WLAN:
echo inet [dirección IP] [mascara] > etc/hostname.[nombre]

Otro comando que se puede emplear es:

LAN
cat hostname.[nombre]
inet [dirección IP] [mascara]

WLAN
cat hostname.[nombre]
inet [dirección IP] [mascara]

Verificar el funcionamiento:
ifconfig

Configurar el gateway, para lo cual se edita el archivo /etc/mygate y se define la IP de enlace o puerta de enlace
cat /etc/mygate
[dirección IP]

Posteriormente se debe definir el reenvío de paquetes entre cada una de las tarjetas de red, para lo cual se modifica el archivo */etc/sysctl.conf*, editando la variable *net.inet.ip.forwarding y* colocando un *"1"* [21], así*:*

cat /etc/sysctl.conf | grep "net.inet.ip.forwarding"
net.inet.ip.forwarding=1

B. Definir reglas.

Una vez se cuente con la adecuada planificación se deben definir las reglas, especificando qué se permite o qué se deniega acorde a las políticas establecidas, esta reglas deben ser muy específicas respecto de cuál va a ser el tipo de tráfico a filtrar, donde se deben tener en cuenta las direcciones IP de origen, IP de destino, protocolos. Así mismo es primordial definir el filtro de los puertos hacia el exterior de la red y el tipo de contenido a ser filtrado tanto entrante como el que sale de la red, la siguiente es la estructura de una regla [22] para pf de OpenBSD:

action [direction] [log] [quick] [on interface] [af] [proto protocol] [from src_addr [port src_port]] [to dst_addr [port dst_port] [flags tcp_flags] [state]

1) **Action**: Corresponde a la acción a ejecutar con el paquete y se emplean las siguientes: **pass** para permitir el paso del paquete y **block** para bloquear o rechazar el paquete acorde a lo establecido en las políticas.
2) **Direction:** Especifica la dirección del paquete **in** entrante **out** saliente de la interface de red.
3) **Log**: Se realiza un registro de log del paquete, en caso de que se requiera generar un registro log de todos los paquetes se emplea el comando **log all.**
4) **Quick**: En caso de que a un paquete se haya asignado el comando **quick**, el cual es considerado como la última regla y se realiza la acción específica.
5) **Interface**: Hace relación a la interfaz de red o grupo de interfaces de red por donde circula el paquete.
6) **Af**: Hace referencia a la familia de direcciones de red, ya sean intet4 para IPv4 o inet6 para IPv6.
7) **Protocol**: Determina el protocolo de red de la capa 4 como lo son: tcp, udp, icmp, icmp6; o nombres validos de protocolo de *etc/protocols*, conjunto de protocolos empleando una lista.
8) **src_addr,dst_addr**: Corresponde a las direcciones IP de origen *src addr* y destino *dst addr*; las direcciones pueden ser: una dirección, bloque de direcciones, nombres de dominio, nombre de interfaz de red, entre otras formas de especificar las direcciones
9) **src_port, dst_port**: Hace relación a los puertos de origen src port y los puertos de destino dst port, estos se pueden definir o especificar como: número del puerto de 1 a 65535, conjunto de puertos creados en una lista, rangos empleando operadores (<,>,!=,><, etc.)
10) **tcp_flags**: Se refiere a los indicadores o flags del protocolo TCP, los cuales se especifican como **check.**
11) **mask:** Dentro de los flag están: **SYN** y **ACK.**
12) **state:** Determina la información del estado del paquete la cual es almacenada para realizar la comparación con las reglas, tales como:

- non state: Se emplea en TCP, UDP e ICMP. OpenBSD pf, no realiza el rastreo de la conexión, para TCP, se requieren **flags any**.
- keep state: Es la función predeterminada en las reglas del filtro de paquetes funciona TCP, UDP e ICMP.
- Modulate state: Funciona solo con TCP.
- *Synproxy state:* Trabaja únicamente con conexiones TCP entrantes, previne o protege los servidores de inundaciones TCP SYN falsas. En esta opción se incluyen las opciones **keep state** y **modulate state.**

A continuación, se aprecia un ejemplo [23] de la definición de una regla de filtrado, donde se aprecian los componentes básicos de la sintaxis de una regla.

Pass TCP traffic in to the web server running on the OpenBSD machine.
pass in on egress proto tcp from any to egress port www

C. Listas

Pf de OpenBSD permite especificar criterios similares dentro de una misma regla, como: varios protocolos, números de puertos, direcciones de origen o destino, entre otros criterios; con lo cual es más fácil

al momento de crear una regla para múltiples direcciones o criterios empleados, la lista se define dentro de corchetes {}, ejemplo: {192.168.1.78, 127.12.3.1, 10.10.3.12}, las comas "," dentro una lista son opcionales; es de mencionar que dentro de una regla se pueden incluir varias listas, lo que permite administrar de forma más eficiente la escritura de las reglas.

*pass in on eth1 proto **{ tcp udp }** from **{ 10.18.1.1, \ 10.5.32.6 }** to any port **{ telnet ssh }***

D. Macros

Son variables que pueden ser definidas por los usuarios, las cuales pueden contener, puestos, direcciones, interfaces, etc. El empleo de macros reduce la complejidad de las reglas o un grupo de reglas de filtrado de paquetes. Para designarlas no se deben emplear palabras reservadas (*out, pass, block*, etc); para nombrar una variable macro de bebe iniciar con una letra, se pueden usar números, letras guion bajo, ejemplo:

red_int = "lmx1"

Al memento de generar la regla y cuando se hace referencia al macro se debe preceder del carácter "**$**"

pass in on $**red_int** from any to any

E. Tablas.

Generalmente empleadas para agrupar o contener grupos de direcciones IPv4 o IPv6, se asignan dentro de los símbolos de "< >". Una de las ventajas de utilizar las tablas es el menor consumo de memoria durante las búsquedas, se pueden emplear en: dirección de origen y/o de destino en reglas de filtrado, scrub, NAT y redireccionamiento, traducción de direcciones en reglas de NAT, entre otras. Para crear una tabla se acede a ***pf.conf*** y se crea la tabla respetiva la cual puede constar de dos tipos de atributos: ***co*nst** donde no se puede cambiar el contenido y **persist** con lo cual el núcleo del sistema conserva el contenido de la tabla, la cual se borrará una vez se ejecute todas las reglas, en el siguiente ejemplo observamos la creación de una tabla:

table <red1> { 192.168.1.0/24 }
table <red2> const { 192.168.2.0/16, 172.167.1.0/12, \10.0.0.0/8 }
table <introspam> persist

Finalmente, una vez definidos los conceptos básicos de filtrado de paquetes con OpenBSD, se hace una breve explicación acerca de la activación, la cual se encuentra contenida en el fichero ubicado en */etc/rc.conf* y configuración, que se encuentra dentro de /etc/pf.conf.

Para la activación del filtrado de paquetes (pf) se modifica el fichero **/etc/rc.conf,** en la línea

pf=YES

También se pueden emplear los comandos de **pfctl**:

pfctl -e # activa pf
pfctl -d # desactiva pf

Luego se realiza el reinicio del sistema, con estos comandos no se realiza la carga del conjunto de reglas establecidas.

Para generar la configuración del paquete de reglas se accede al sistema de archivos y ubicamos el fichero */etc/pf.conf***,** este es un fichero de texto cargado e interpretado por **pfctl**.

El fichero de configuración consta de siete partes [23], las cuales excepción de las macros y tablas, cada sección debería aparecer en el mismo orden:

1) **Macros.**
2) **Tablas**.
3) **Opciones:** Empleadas para el funcionamiento del pf, como el tiempo de expiración de conexiones, tiempos de espera, acciones para paquetes bloqueados, y criterios de **aggressive** corta conexiones no activas y **conservative**, conserva las conexiones, entre otros.
4) **Normalización (Scrub):** Permite establecer reglas para reprocesamiento de paquetes de normalización y desfragmentación, siendo muy útil declarar que los paquetes no estén fragmentados antes de ingresar a la red.
5) **Formación de colas:** Provee control del ancho de banda y priorización de paquetes.
6) **Traducción de direcciones de red NAT:** Se pueden establecer las reglas que indican como mapear/traducir las peticiones salientes de la red local hacia internet o viceversa, así mismo permite el redireccionamiento de paquetes.
7) **Reglas de filtrado:** Empleado para filtrado selectivo o el bloqueo de paquetes, nos permite establecer las reglas de filtrado. La manera en que pf evalúa estas reglas de la primera a la última, la última regla coincidente es la aplicada. Una regla que tenga la

palabra clave **quick** marcada para un paquete, automáticamente se aplica esa regla y no se continua el recorrido con el listado de reglas

A continuación, se presenta un ejemplo [23] del conjunto de reglas, en las cuales se incluyen listas, macros, opciones NAT y filtrado de paquetes.

```
# Macros y listas pueden combinarse
int if=``em1"
tcp services=``f 22, 113 g"
udp services=``f domain g"
icmp types=``echoreq"

# Opciones
set block-policy return
set loginterface em0
set skip on lo

# NAT
match out on egress inet from !(egress) to any nat-to (egress:0)

# Filtrado – se bloquea tráfico en todas direcciones.
block in log
pass out quick
antispoof quick for f lo $int if g #Antispoofing

# Permitimos paso a protocolos y puertos autorizados
pass in on egress inet proto tcp from any to port $tcp services
pass proto udp to port $udp services
pass in inet proto icmp all icmp-type $icmp types #ping
pass in on $int if # confiamos en tráfico de interfaz interno
```

VII. INSTALACIÓN DE OPENBSD

Para el procedimiento de instalación del software OpenBSD y la configuración del sistema de filtrado de paquetes (pf), se opta por emplear la plataforma de desarrollo BeagleBone Black y un sistema de cómputo de bajos recursos tecnológicos de la marca Acer, modelo Aspire One; para lo cual, la aplicación propuesta comprende el uso de dos tarjetas de red, una de las cuales se encuentra dispuesta hacia la red de internet y la otra configurada a red LAN. En figura 3 se detalla el esquemático de la configuración a emplear.

Se realiza la aclaración que, a fín de no extender este documento y el cual no pretende ser un tutorial de instalación, no se detallan cada uno de los procesos que se requieren durante la instalación, por lo cual se dan las pautas generales para puesta en funcionamiento del sistema operativo OpenBSD.

Una vez instalados los sistemas de manera general, se deben realizar los ajustes pertinentes en cuanto a: configuración de tarjetas de red, inicio del filtrado de paquetes, fijación de reglas de filtrado, descarga de repositorios o paquetes.

Fig. 3. Esquema de configuración

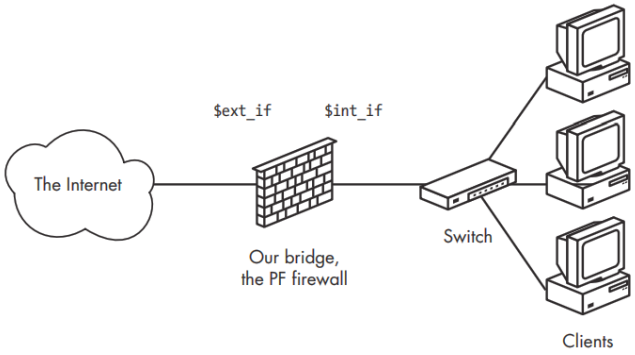

Fig. 3: Esquema para la configuración elegida. Fuente: Book of PF, 3rd Edition. A No-Nonsense Guide to the OpenBSD firewall, Peter N. M. Hansteen October 2014, 248 pp.

A. Instalación en BeagleBone Black.

Para proceder a la instalación se realiza la búsqueda en la página oficial del proyecto OpenBSD *https://www.openbsd.org* [24], en la sección "*Download*", se procede a la descarga de sistema, así mismo se encuentran las instrucciones para la instalación, dependiendo del sistema escogido, para el caso se realizó la descarga del archivo armv7 en "*Index of /pub/OpenBSD/6.3/armv7/ miniroot-am335x-63.fs"* [25]. Debido al auge en la propagación de software malicioso, la fundación OpenBSD recomienda realizar la verificación del software descargado, comprobando la integridad del archivo con sha256, el cual se encuentra disponible en la página de *openbsd.org*, lo que permite que podamos tener la seguridad de que el archivo de instalación descargado no fue alterado o infectado con malware. Previa a la instalación en la BeagleBone Black, se hace necesario contar con los siguientes elementos:

- BeagleBone Black.
- MicroSD mínimo 4GB.
- Cable USB a serial RS232 de 3.3 voltios (TTL).
- Cable Ethernet con conexión a internet.
- Tarjeta red USB a rj45.

Se debe tener especial cuidado en la conexión de los cables entre la BeagleBone Black y el cable serial [26]; los pines empleados en la BeagleBone son: ground (1), RX (4) y TX (5) del puerto serial debug, donde deben estar conectados el pin de ground, los pines RX y TX del cable serial, estos dos últimos de forma invertida así: RX a TX (BeagleBone Black) y de TX a RX (BeagleBone Black). Para la instalación del software en la memoria se puede emplear el comando: *dd if=miniroot-am335x-63.fs of=/dev/rsd1c,* donde sd1 corresponde al nombre del dispositivo, el cual puede

cambiar según la lista de dispositivos, con el comando *fdisk –l* se pueden listar para determinar el nombre de la unidad y realizar la instalación; una vez conectado el dispositivo USB al equipo, se procede a conectar en consola y realizar la configuración del puerto serial, luego se energiza la placa BeagleBone Black y se mantiene presionado el botón de power, lo que hace que el dispositivo arranque desde la memoria *microSD* e inicie el proceso de instalación del sistema operativo OpenBSD [25], para el caso se seleccionó la opción **(I)nstall,** en la figura 4 se aprecian las diferentes opciones de instalación, una vez inicia la instalación, el software solicita parámetros como: interfaces, DHCP, usuario y password, configuración de paquetes de descarga, entre otras opciones, las cuales se configuran acorde a los requerimientos del usuario.

Fig. 4. Pantallazo opciones de instalación

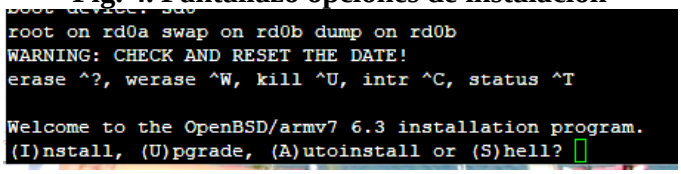

Fig. 4: Opciones de instalación en el dispositivo BeagleBone Black. Fuente el autor.

Finalizada la instalación se procede a realizar las configuraciones de cada una de las redes e implementar las diferentes reglas de filtrado acorde a las necesidades o políticas de los usurarios [20] y [22].

B. Instalación en equipo Acer Aspire One.

Para este caso se seleccionó un computador portátil, con bajas prestaciones tecnológicas, como son:

- Procesador Intel Atom de 1.66Ghz. Figura 5.
- Memoria RAM 1024 Mb. Figura 6.
- Un puerto Ethernet 10/100 Mbps, para la instalación de OpenBSD, se adiciona un puerto conversor USB a Rj45.
- Disco duro 80Gb.

Se aprecia la configuración del equipo seleccionado para la instalación del OpenBSD pf.

Fig. 5. Imagen configuración procesador

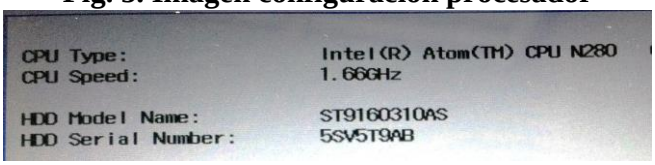

Fig. 5: Imagen obtenida, donde se detalla el tipo y velocidad del procesador del equipo. Fuente el autor.

Fig. 6. Pantallazo cantidad de memoria RAM

Fig. 6: Imagen obtenida, donde se detalla la cantidad de memoria RAM del equipo. Fuente el autor

Para la instalación se realiza la búsqueda en la página oficial del proyecto OpenBSD *https://www.openbsd.org* [24], en la sección "*Download*", se realiza la descarga de sistema, para este equipo de cómputo se toma la opción del archivo "*install63.fs* [*i386]*", o el archivo "*install63.iso [i386*]; es de mencionar que otra de las opciones para la descarga del sistema OpenBSD es ingresar al servidor FTP http://mirrors.unb.br/pub/OpenBSD/6.3/i386/, donde se halla el archivo de instalación "*install63.iso"* y luego de verificada la integridad descarga se procede a la instalación [27].

Como procedimiento adicional en el sistema OpenBSD de escritorio se procede a realizar modificación para automatizar la descarga de paquetes necesarios para la configuración y personalización del sistema, para lo cual se emplea el siguiente comando:

export PKG_PATH=https://ftp.openbsd.org/pub/OpenBSD/6.3/packages/$(uname -m)

Luego para que este comando quede permanente en el sistema con el administrador root se ingresa el siguiente comando:

echo export PKG_PATH=https://ftp.openbsd.org/pub/OpenBSD/6.3/packages/$(uname -m) >> .profile

Con estos comandos ya es posible descargar diferentes paquetes como editores de texto, fondo de escritorio, si se desea la versión gráfica, openoffice, entre otras aplicaciones.

Finalmente se realiza la configuración del filtrado de paquetes acorde al sistema propuesto [20] y [22].

VIII. CONCLUSIONES

Al implementar el filtrado de paquetes como base fundamental de la seguridad de la información en hogares y empresas, estamos contribuyendo o aportando nuestro grano de arena en la consecución del

objetivo general planteado en el documento CONPES 3854 "Política Nacional de Seguridad Digital" [28] y en especial con lo relacionado con mitigar los riesgos informáticos.

En el documento CONPES 3701 "Lineamientos de Política para Ciberseguridad y Ciberdefensa", se plantea que: "*es de vital importancia crear conciencia y sensibilizar a la población en todo lo referente a la seguridad de la información"* [29], por lo cual se hace necesario que se capacite y se implementen sistemas de software y hardware de bajo costo para los ciudadanos, para la instalación de filtrado de paquetes, como primer bastión de la seguridad en las redes de información.

Es posible aprovechar la infraestructura creada por "*Computadores para Educar*" [2], con el fin de generar un plan para capacitar en riesgos informáticos e implementar el filtrado de paquetes en las instituciones a las cuales llega el programa y aprovechando los equipos en reuso u obsolescencia tecnológica para tal fin.

La ventaja principal de OpenBSD radica en que se encuentra disponible para múltiples plataformas o arquitecturas, desde dispositivos embebidos como BeagleBone Black, Rasperry Pi, hasta robustos equipos de cómputo e incluso servidores, con lo cual es posible masificar su instalación; así mismo por tratarse de software libre, su implementación no precisa de grandes recursos económicos para la puesta en funcionamiento.

El proceso de instalación y configuración del filtrado de paquetes es sencillo; sin embargo, la parte que precisa de cierto cuidado y conocimiento es lo relacionado con la creación y aplicación de reglas de filtrado, para lo cual en la planeación y generación de las políticas de seguridad se debe tener en cuenta el tipo de tráfico y categorizar cuales son las áreas primordiales a proteger.

El empleo de software de filtrado de paquetes de OpenBSD precisa grandes ventajas, ya que no solo realiza filtrado de paquetes en IPv4 e IPv6, si no, que además, permite balanceo de cargas, implementación de NAT (Network Address Translation), normalización de paquetes y modulación del ancho de banda, entre otros beneficios. Por tanto, lo hace un software robusto a la hora de minimizar y proteger nuestra información.

REFERENCIAS

[1] Ley N°. 1672. Diario oficial de la República de Colombia. No. 48856 del 19 de julio de 2013

[2] Reutilización y puesta en funcionamiento de equipos. Recuperado de: http://www.computadoresparaeducar.gov.co/

[3] "e-waste" (n.d). Recuperado de: http:// www.raee.org.co. 6 de abril de 2018.

[4] Decreto 284 de 2018. Ministerio de Ambiente y Desarrollo Sostenible.

[5] What is BeagleBone Black? Recuperado de: https://beagleboard.org/black.

[6] Karen Scarfone, Paul Hoffman. Guidelines on firewalls and firewall policy 800-41. Recuperado de: https://csrc.nist.gov/publications/detail/sp/800-41/rev-1/final

[7] OpenBSD filtrado de paquetes. Recuperado de https://docstore.mik.ua/manuals/openbsd/faq/pf/es/filter.html.

[8] The IP Network Address Translator (NAT). Network working group. Recuperado de: https://tools.ietf.org/html/rfc1631

[9] Defense advanced research projects agency. Internet Protocol RFC791. Disponible en: http://www.rfc-base.org/txt/rfc-791.txt.

[10] Transmission control protocol. Defense advanced research projects. RFC793. Disponible en: https://tools.ietf.org/html/rfc793.

[11] User datagram protocol. Disponible en: https://www.rfc-editor.org/rfc/rfc768.txt

[12] Ganesh Dutt Sharma. Packet filtering firewall: An introduction. 2010. Recuperado de: http://securityworld.worldiswelcome.com/packet-filtering-firewall-an-introduction.

[13] Mecanismos de filtrado stateless y stateful. Recuperado de: https://ostec.blog/es/seguridad-perimetral/firewall-stateful-stateless.

[14] OpenBSD filtrado de paquetes. Recuperado de: https://docstore.mik.ua/manuals/openbsd/faq/pf/es/filter.html

[15] The OpenBSD foundation - Funding for OpenBSD and related projects. Recuperado de: https://www.openbsdfoundation.org/

[16] Only two remote holes in the default install, in a heck of a long time. Disponible en: https://www.openbsd.org/

[17] Espíndola Raúl. OpenBSD. Recuperado de: https://raulespinola.wordpress.com/tag/openbsd/

[18] OpenBSD platforms. Disponible en: https://www.openbsd.org/plat.html.

[19] Packet filter firewall and packet processing. Oracle. Disponible en: https://docs.oracle.com/cd/E53394_01/html/E54829/pfovw-proc.html

[20] OpenBSD pf - Packet filtering. Disponible en: https://www.openbsd.org/faq/pf/

[21] Peter N.M. Hansteen. Book of pf. A no-nonsense guide to the OpenBSD. 3rd edition. October 2014, 248 pp.

[22] OpenBSD pf: Filtrado de paquetes. Recuperado de: https://docstore.mik.ua/manuals/openbsd/faq/pf/es/filter.html

[23] Cortafuegos con software libre. Diez años de pf. Master oficial en software libre. Miguel Vidal. Disponible en: http://gsyc.urjc.es/~mvidal 2011.

[24] OpenBSD pf. Download. Disponible en: https://www.openbsd.org/faq/faq4.html#Download.

[25] Installation notes for OpenBSD/armv7 6.3. Disponible en: https://ftp.openbsd.org/pub/OpenBSD/6.3/armv7/INSTALL.armv7

[26] OpenBSD 6.2 on BeagleBone Black. Last edited Wed Jan 3 16:58:05 2018. Disponible en: https://box.matto.nl/openbsd62beagleblack.html

[27] Installation notes for OpenBSD/i386 6.3. Disponible en: https://ftp.openbsd.org/pub/OpenBSD/6.3/i386.

[28] Política nacional de seguridad digital CONPES 3854. 2016. Recuperado de: https://colaboracion.dnp.gov.co/CDT/Conpes/Econ%C3%B3micos/3854.pdf.

[29] Lineamientos de política para ciberseguridad y ciberdefensa. CONPES 3701. 2011. Disponible en: https://colaboracion.dnp.gov.co/CDT/Conpes/Econ%C3%B3micos/3701.pdf

Autor:
Luis Jesús Sepúlveda Manrique. Tecnólogo en Electrónica, Ingeniero en Telecomunicaciones, actualmente culminando estudios de Especialización en Seguridad Informática en la Universidad Piloto de Colombia.